Logistic Regression

Nipun Batra

IIT Gandhinagar

July 30, 2025

Aim: Probability(Tomatoes | Radius) ? or

Aim: Probability(Tomatoes $|$ Radius) ? or

$\boxed{\text{More generally, P}(y = 1|\mathbf{X} = \mathbf{x})\text{?}}$

Generally,

$$P(y = 1|\mathbf{x}) = \mathbf{X}\boldsymbol{\theta}$$

$\sigma(z) \to 1$

$\sigma(z) \to 1$
$z \to -\infty$

$\sigma(z) \to 1$
$z \to -\infty$
$\sigma(z) \to 0$

$\sigma(z) \to 1$
$z \to -\infty$
$\sigma(z) \to 0$
$z = 0$

$$\sigma(z) \to 1$$
$$z \to -\infty$$
$$\sigma(z) \to 0$$
$$z = 0$$
$$\sigma(z) = 0.5$$

$$P(y=0|X) = 1 - P(y=1|X) = 1 - \frac{1}{1+e^{-\mathbf{X}\boldsymbol{\theta}}} = \frac{e^{-\mathbf{X}\boldsymbol{\theta}}}{1+e^{-\mathbf{X}\boldsymbol{\theta}}}$$

$$P(y=0|X) = 1 - P(y=1|X) = 1 - \frac{1}{1+e^{-\mathbf{X}\boldsymbol{\theta}}} = \frac{e^{-\mathbf{X}\boldsymbol{\theta}}}{1+e^{-\mathbf{X}\boldsymbol{\theta}}}$$

$$\therefore \frac{P(y=1|X)}{1-P(y=1|X)} = e^{\mathbf{X}\boldsymbol{\theta}} \implies \mathbf{X}\boldsymbol{\theta} = \log\frac{P(y=1|X)}{1-P(y=1|X)}$$

Why? Squared loss + sigmoid creates non-convex surface:

- Sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$ is non-linear

Why? Squared loss + sigmoid creates non-convex surface:

- Sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$ is non-linear
- Composition $(\sigma(\mathbf{X}\boldsymbol{\theta}) - y)^2$ has multiple local minima

Why? Squared loss + sigmoid creates non-convex surface:

- Sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$ is non-linear
- Composition $(\sigma(\mathbf{X}\boldsymbol{\theta}) - y)^2$ has multiple local minima

Why? Squared loss + sigmoid creates non-convex surface:

- Sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$ is non-linear
- Composition $(\sigma(\mathbf{X}\boldsymbol{\theta}) - y)^2$ has multiple local minima
- No guarantee gradient descent finds global optimum

Why? Squared loss + sigmoid creates non-convex surface:

- Sigmoid $\sigma(z) = \frac{1}{1+e^{-z}}$ is non-linear
- Composition $(\sigma(\mathbf{X}\boldsymbol{\theta}) - y)^2$ has multiple local minima
- No guarantee gradient descent finds global optimum
- This is why we need cross-entropy loss instead!

This cost function is called cross-entropy.

This cost function is called cross-entropy.
Why?

What is the interpretation of the cost function?

What is the interpretation of the cost function?
Let us try to write the cost function for a single example:

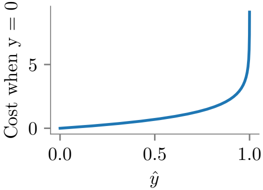
What is the interpretation of the cost function?
Let us try to write the cost function for a single example:

$$J(\theta) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$$

What is the interpretation of the cost function?
Let us try to write the cost function for a single example:

$$J(\theta) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$$

First, assume y_i is 0, then if $\hat{y}_i$ is 0, the loss is 0; but, if $\hat{y}_i$ is 1, the loss tends towards infinity!



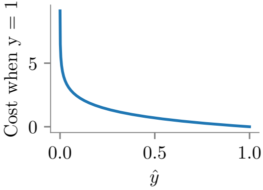
What is the interpretation of the cost function?

$$J(\theta) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$$

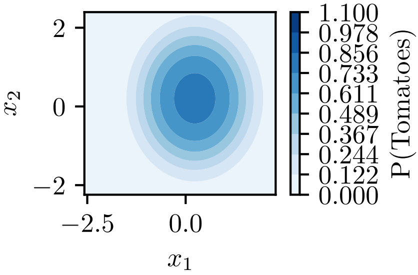
What is the interpretation of the cost function?

$$J(\theta) = -y_i \log \hat{y}_i - (1 - y_i) \log(1 - \hat{y}_i)$$

Now, assume y_i is 1, then if $\hat{y}_i$ is 0, the loss is huge; but, if $\hat{y}_i$ is 1, the loss is zero!



Bias!



2
0
−2
−2.5
0.0
x_1
x_2
1.100
0.978
0.856
0.733
0.611
0.489
0.367
0.244
0.122
0.000
P(Tomatoes)

How would you learn a classifier? Or, how would you expect the classifier to learn decision boundaries?

1. Use one-vs.-all on Binary Logistic Regression

1. Use one-vs.-all on Binary Logistic Regression
2. Use one-vs.-one on Binary Logistic Regression

1. Use one-vs.-all on Binary Logistic Regression
2. Use one-vs.-one on Binary Logistic Regression
3. Extend Binary Logistic Regression to Multi-Class Logistic Regression

1. Learn P(setosa (class 1)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_1)$

1. Learn P(setosa (class 1)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_1)$
2. P(versicolor (class 2)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_2)$

1. Learn P(setosa (class 1)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_1)$
2. P(versicolor (class 2)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_2)$
3. P(virginica (class 3)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_3)$

1. Learn P(setosa (class 1)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_1)$
2. P(versicolor (class 2)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_2)$
3. P(virginica (class 3)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_3)$
4. Goal: Learn $\theta_i \forall i \in \{1, 2, 3\}$

1. Learn P(setosa (class 1)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_1)$
2. P(versicolor (class 2)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_2)$
3. P(virginica (class 3)) = $\mathcal{F}(\mathbf{X}\boldsymbol{\theta}_3)$
4. Goal: Learn $\theta_i \forall i \in \{1, 2, 3\}$
5. Question: What could be an $\mathcal{F}$?

1. Question: What could be an $\mathcal{F}$?

1. Question: What could be an $\mathcal{F}$?
2. Property: $\sum_{i=1}^{3} \mathcal{F}(\mathbf{X}\boldsymbol{\theta}_i) = 1$

1. Question: What could be an $\mathcal{F}$?
2. Property: $\sum_{i=1}^{3} \mathcal{F}(\mathbf{X}\boldsymbol{\theta}_i) = 1$
3. Also $\mathcal{F}(z) \in [0, 1]$

1. Question: What could be an $\mathcal{F}$?
2. Property: $\sum_{i=1}^{3} \mathcal{F}(\mathbf{X}\boldsymbol{\theta}_i) = 1$
3. Also $\mathcal{F}(z) \in [0, 1]$
4. Also, $\mathcal{F}(z)$ has squashing proprties: $R \mapsto [0, 1]$

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$
= $-(0 \times \log(0.1) + 1 \times \log(0.8) + 0 \times \log(0.1))$

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$
= $-(0 \times \log(0.1) + 1 \times \log(0.8) + 0 \times \log(0.1))$
Tends to zero

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$
= $-(0 \times \log(0.1) + 1 \times \log(0.4) + 0 \times \log(0.1))$

Let us calculate $-\sum_{k=1}^{3} y_i^k \log \hat{y}_i^k$
= $-(0 \times \log(0.1) + 1 \times \log(0.4) + 0 \times \log(0.1))$
High number! Huge penalty for misclassification!

More generally,

More generally,

$$J(\theta) = -\left\{ \sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right\}$$

More generally,

$$J(\theta) = -\left\{ \sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right\}$$

$$J(\theta) = -\left\{ \sum_{i=1}^{N} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \right\}$$

Extend to K-class:

$$J(\theta) = -\left\{ \sum_{i=1}^{N} \sum_{k=1}^{K} y_i^k \log(\hat{y}_i^k) \right\}$$

What is the key difference between sigmoid and softmax functions?

What is the key difference between sigmoid and softmax functions?

What is the key difference between sigmoid and softmax functions?

Why do we use cross-entropy loss instead of squared error?

What is the key difference between sigmoid and softmax functions?

Why do we use cross-entropy loss instead of squared error?

What is the key difference between sigmoid and softmax functions?

Why do we use cross-entropy loss instead of squared error?

How does regularization help in logistic regression?

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function
- **Linear Decision Boundary**: Creates linear separation in feature space

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function
- **Linear Decision Boundary**: Creates linear separation in feature space

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function
- **Linear Decision Boundary**: Creates linear separation in feature space
- **Maximum Likelihood**: Optimized using gradient-based methods

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function
- **Linear Decision Boundary**: Creates linear separation in feature space
- **Maximum Likelihood**: Optimized using gradient-based methods

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function
- **Linear Decision Boundary**: Creates linear separation in feature space
- **Maximum Likelihood**: Optimized using gradient-based methods
- **Cross-Entropy Loss**: Appropriate for classification problems

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function
- **Linear Decision Boundary**: Creates linear separation in feature space
- **Maximum Likelihood**: Optimized using gradient-based methods
- **Cross-Entropy Loss**: Appropriate for classification problems

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function
- **Linear Decision Boundary**: Creates linear separation in feature space
- **Maximum Likelihood**: Optimized using gradient-based methods
- **Cross-Entropy Loss**: Appropriate for classification problems
- **No Closed Form**: Requires iterative optimization (gradient descent)

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function
- **Linear Decision Boundary**: Creates linear separation in feature space
- **Maximum Likelihood**: Optimized using gradient-based methods
- **Cross-Entropy Loss**: Appropriate for classification problems
- **No Closed Form**: Requires iterative optimization (gradient descent)

Key Takeaways

- **Probabilistic Model**: Outputs probabilities via sigmoid function
- **Linear Decision Boundary**: Creates linear separation in feature space
- **Maximum Likelihood**: Optimized using gradient-based methods
- **Cross-Entropy Loss**: Appropriate for classification problems
- **No Closed Form**: Requires iterative optimization (gradient descent)
- **Regularization**: L1/L2 help prevent overfitting